REMARKS

The Office action has been carefully considered. The Office action rejected claims 1, 3, 4, 8-12 and 16 under 35 U.S.C. § 102(b) as being anticipated by U.S. Patent No. 5,805,899 to Evans et al. ("Evans"). Additionally, the Office action rejected claims 2, 5-7, 13-14, 16-18 and 20-26 under 35 U.S.C. § 103(a) as being unpatentable over Evans in view of U.S. Patent No. 6,560,614 to Barboy et al. ("Barboy"). Further, the Office action rejected claim 19 under 35 U.S.C. § 103(a) as being unpatentable over Evans in view of Barboy and in further view of U.S. Patent No. 4,809,170 to Leblang et al. ("Leblang"). Still further, the Office action rejected claims 28-37 under 35 U.S.C. § 103(a) as being unpatentable over Evans. Finally, the Office action rejected claims 15 and 20 under 35 U.S.C. §112, second paragraph as being indefinite. Applicants have amended claims 15 and 20 to obviate the §112, second paragraph rejection. Regarding the rejections based on §102 and §103, applicants respectfully disagree and traverse.

By present amendment, claims 1 and 17 have been amended for clarification and not in view of the prior art. Applicants submit that the claims as filed were patentable over the prior art of record, and that the amendments herein are for purposes of clarifying the claims and/or for expediting allowance of the claims and not for reasons related to patentability. Reconsideration is respectfully requested.

Applicants thank the Examiner for the interview held (by telephone) on March 15, 2004. During the interview, the Examiner and applicants' attorney discussed the claims with respect to the prior art. The essence of applicants' position is incorporated in the remarks below.

Prior to discussing reasons why applicants believe that the claims in this application are clearly allowable in view of the teachings of the cited and applied references, a brief description of the present invention is presented.

The present invention is directed to a method, system, and infrastructure that allow an application to run with specified versions of assemblies bound thereto, while allowing the application author, assembly publisher, and/or system administrator to change the originally-specified version as desired.

Applications in today's software environment typically use components (which may be objects) that are dynamically linked to the applications during runtime. In this manner, a single component may be used by many different applications and the need for several copies of one component is eliminated. However, as components become updated (that is, a newer version of the component replaces an old version), applications which previously had the old version dynamically linked may no longer operate correctly with the new version. Thus, either all components need to be backwards compatible or the older versions of the components need to still be available for applications that require the older version components.

Components are often packaged with other components to form an assembly. Assemblies, like individual components, also have versioning information that indicates when the assembly was last updated. In this manner, a group of components may be updated, but only the version of the assembly needs to be modified to indicate a newer assembly version. Applications can then be bound (linked) to one assembly of components, rather than many bindings to several different components.

In one embodiment of the present invention, a manifest stores configuration information about each application that indicates which assembly version is required for the application to run. As such, only the assembly version need be stored in the configuration information in the manifest for each application as opposed to many versions of several components. Further, the manifest may store different kinds of configurations, such as the original publisher's configuration which includes only the version information of the assembly at the time the publisher released the application, an administrator's configuration which includes certain versions that allow particular activities that an administrator may require, or the application's current configuration that includes all version updates.

Note that the above description is for example and informational purposes only, and should not be used to interpret the claims, which are discussed below.

§102(b) Rejections

Turning to the claims, claim 1, as amended, recites a computer-implemented method, comprising receiving a request corresponding to binding at least one shared assembly to executable code, and interpreting configuration information corresponding to the at least one shared assembly to determine a version of a shared assembly to bind to the executable code, wherein the configuration information is separate from the shared assembly.

The Office action rejected claim 1 as unpatentable over Evans. Specifically, the Office action contends that Evans teaches receiving a request corresponding to binding at least one shared assembly to executable code. Column 1, lines 23-24 of Evans are referenced. Further, the Office action contends that Evans teaches interpreting configuration information to determine a version of a shared assembly to bind to the executable code, wherein the configuration information is separate from the shared assembly. Column 14, lines 57-61 of Evans are referenced. Applicants respectfully disagree.

Evans teaches, generally, a single-tiered versioning system for individual software objects. In this versioning system, a mapfile is used to a store version directive (versioning information) about each of several objects that may be shared. See column 5, line 66 to column 6, line 3. When a link-editor is directed to produce a shared object during a build-time operation, a particular version of the object is identified in the mapfile and then retrieved from a cache of relocatable objects. As such, the new object created by the link-editor for this particular application is said to be a versioned object because the object was dynamically linked with the version indicated by the application itself. See column 5, lines 7-32 of Evans.

For example, an application may require an object called ABC. Further, the application may even require a particular version of ABC, such as ABC 1.2.3. When the application is executed, the link-editor is called upon to instantiate the required objects. Thus, the link-editor first looks to the mapfile to determine the available versions of the requested object ABC. If the particular version is found, the link-editor instantiates the object ABC 1.2.3 and this object is referred to as a versioned object because the version information is part of the instantiation.

The recitations of claim 1, however, are directed toward a two-tiered versioning system. That is, the version of the objects themselves are maintained in a manifest as well as the version of the assemblies of the objects. Further, each assembly may have configuration information associated with it that is indicative of a particular operating preference, *e.g.*, an administrator's configuration or a publisher's configuration. As such, not only are the individual objects identified by one aspect of a versioning system, the assemblies of objects are also identified by another aspect of the versioning system. Additionally, specific assembly versions can be further identified by configuration information.

As such, Evans does not teach the recitations of claim 1. First, Evans teaches a versioning system directed toward individual objects. This is not the same as receiving a request corresponding to binding at least one shared assembly to executable code. That is, individual objects are not the same as an assembly of objects. Next, Evans teaches a versioning system that calls for individual linking of individual objects to executable code. Evans, however, does not teach nor show any appreciation of using assemblies of versioned objects to link to executable code. That is, Evans does not teach determining a version of a shared assembly to link to executable code as generally recited in claim 1. Finally, because Evans does not teach nor show any appreciation of using assemblies of versioned objects to link to executable code, Evans cannot possibly teach configuration information corresponding to the assembly as recited in claim 1.

Applicants submit that claim 1, as amended, is allowable over the prior art of record for at least the foregoing reasons.

Applicants respectfully submit that dependent claims 3, 4, 8-12 and 16, by similar analysis, are allowable. Each of these claims depends either directly or indirectly from claim 1 and consequently includes the recitations of independent claim 1. As discussed above, Evans fails to disclose the recitations of claim 1 and therefore these claims are also allowable over the prior art of record. In addition to the recitations of claim 1 noted above, each of these dependent claims includes additional patentable elements.

For example, claim 8 recites associating at least one assembly version with a publisher configuration having redirection information therein, wherein interpreting configuration information includes interpreting the publisher configuration. As discussed above, Evans does not teach nor show any appreciation of using assemblies of versioned objects to link to executable code, let alone configurations of assemblies. Therefore, Evans cannot possibly teach a publisher configuration having redirection information as recited in claim 8.

As another example, claim 10 recites interpreting configuration information includes interpreting an administrator configuration. Again, Evans does not teach nor show any appreciation of using assemblies of versioned objects to link to executable code, let alone configurations of assemblies. Therefore, Evans cannot possibly teach an administrator configuration as recited in claim 10.

For at least these reasons, applicants submit that claims 3, 4 8-12, and 16 are allowable over the prior art of record.

§103(a) Rejections

By law, in order to establish *prima facie* obviousness of a claimed invention, all of the claim limitations must be taught or suggested by the prior art. *In re Royka*, 490 F.2d 981, 180 USPQ 580 (CCPA 1974). In addition, "all words in a claim must be considered in judging the patentability of that claim against the prior art." *In re Wilson*, 424 F.2d 1382, 1385, 165 USPQ 494, 496 (CCPA 1970). Further, if prior art, in any material respect teaches away from the claimed invention, the art cannot be used to support an obviousness rejection. *In re Geisler*, 116 F.3d 1465, 1471, 43 USPQ2d 1362, 1366 (Fed Cir. 1997). As discussed in greater detail below, the claims of the present invention are thus clearly patentable over the teachings of the cited and applied references as a matter of law.

The Office action rejected claims 2, 5-7, 13-14 and 16 as being unpatentable over Evans in view of Barboy. These claims depend either directly or indirectly on claim 1 and, therefore, include each element of claim 1. As discussed above, Evans fails to disclose each and every element of claim 1. Furthermore, Barboy also fails to teach the recitations of claim 1. Nowhere in Barboy can there be found any disclosure of a computer-implemented method, comprising receiving a request corresponding to binding at least one shared assembly to executable code, and interpreting configuration information corresponding to the at least one shared assembly to determine a version of a shared assembly to bind to the executable code, wherein the configuration information is separate from the shared assembly, as recited by claim 1. Barboy instead relates to a software updating system that can be used to provide the newest update of a software program to a user without their knowledge. Neither Evans nor Barboy, whether considered alone or in any permissible combination, disclose each and every element of claim 1, and therefore, do not make obvious, in any permissible combination, dependent claims 2, 5-7, 13-14 and 16 which also individually include every element of claim 1. Further, applicants submit that these claims recite additional limitations that are not made obvious by Evan or Barboy.

For example, claim 2 recites that interpreting configuration information includes redirecting one assembly version to another assembly version. The Office action acknowledges that Evans does not teach redirecting one assembly version to another assembly version. However, the Office action contends that Barboy teaches redirecting calls for an out-of-date software program to a new software program and concludes that the recitations of claim 2 would have been obvious to a person skilled in the art at the time that the invention was made since the teachings of Barboy allow users to access a newer software version without having knowledge that the older version has been updated. Applicants respectfully disagree.

Barboy teaches, generally, a software updating system that can be used to provide the newest update of a software program to a user without their knowledge. However, the newest version of a software program is not the same as a different version of an assembly of objects as recited in claim 2. The present invention is not directed toward providing the newest form of an application, but rather providing a specific configuration of objects in an assembly of objects to an application. Barboy specifically *teaches away* from this as older versions of applications are not available to the user since the user is unaware of the redirection.

For at least these reasons, applicants submit that claims 2 5-7, 13-14 and 16 are patentable over the prior art of record.

Turning to the next independent claim, claim 17, as amended, recites a system in a computing environment, comprising a manifest including information that specifies a dependency of executable code on an identified version of a shared assembly, a configuration corresponding to the shared assembly including information that redirects at least one version of a shared assembly to another version of that shared assembly, and a binding mechanism configured to receive a request directed to execute the executable code, to select the identified version of the shared assembly from the manifest, and to interpret the configuration to determine whether to redirect the identified version in the manifest to another version identified in the configuration.

The Office action rejected claim 17 as being unpatentable over Evans in view of Barboy. Specifically, the Office contends that Evans teaches a manifest including information that specifies a dependency of executable code on an identified version of a shared assembly. Column 12, lines 50-53 of Evans are referenced. Further, the Office action contends that Evans teaches a configuration corresponding to the shared assembly. Figure 2(a) of Evans is referenced. Further yet, the Office action contends that Evans teaches a binding mechanism configured to receive a request directed to execute the executable code, to select the identified version of the shared assembly from the manifest. Again, Figure 2(a) of Evans is referenced. Finally, the Office action concedes that Evans does not teach the shared assembly including information that redirects at least one version of a shared assembly to another version of that shared assembly or a binding mechanism configured to redirect the identified version in the manifest to another version identified in the configuration. However, the Office action contends that Barboy teaches these recitations and that the combination of Evans and Barboy would render obvious the recitations of claim 17. Applicants respectfully disagree.

As discussed above, Barboy teaches, generally, a software updating system that can be used to provide the newest update of a software program to a user without their knowledge. However, providing the newest version of a software program is not the same as a providing different versions of an assembly of objects in different configurations. The present invention is not directed toward providing the newest form of an application, but rather providing a specific configuration of objects in an assembly of objects to an application. Barboy *specifically teaches* away from the recitations of claim 17 as older versions of applications are not available to the user since the user is unaware of the redirection. Further, as discussed above, Evans does not teach nor show any appreciation of using assemblies of versioned objects to link to executable code, let alone configurations of assemblies. Significantly, there is no teaching or even any appreciation of assemblies in either the Evans reference or the Barboy reference. Neither Evan nor Barboy, whether considered alone or in any permissible combination, disclose or suggest each and every element of claim 17, and, therefore, the references do not make obvious, in any permissible combination, claim 17.

The Office action rejected claims 18-27 as being unpatentable over Evans in view of either Barboy or Leblang. These claims depend either directly or indirectly from claim 17 and, therefore, include each element of claim 17. As discussed above, Evans fails to disclose each and every element of claim 17. Barboy also fails to teach the recitations of claim 17. Furthermore, Leblang also fails to teach the recitations of claim 17. Nowhere in Leblang can there be found any disclosure of a shared assembly including information that redirects at least one version of a shared assembly to another version of that shared assembly or a binding mechanism configured to redirect the identified version in the manifest to another version identified in the configuration. Leblang is instead related to a source versioning system for building different versions of compiled modules. Neither Evans nor Barboy nor Leblang, whether considered alone or in any permissible combination, disclose each and every element of claim 17, and therefore, do not make obvious, in any permissible combination, dependent claims 18-27 which also individually include every element of claim 1.

For at least these reasons, applicants submit that claims 18-27 are patentable over the prior art of record.

Turning to the next independent claim, claim 28 recites a computer-implemented method, comprising receiving a request corresponding to binding a selected version of an assembly to an application program, determining whether publisher configuration information is associated with the assembly, and if so, interpreting information in the publisher configuration to determine whether to bind to the application program a version of the assembly that is different from the selected version, and if so, selecting that different version as the selected version of the assembly for binding to the application program, and determining whether application configuration information is associated with the application, and if so, interpreting the application configuration information to determine whether to bind to the application program a version of the assembly that is different from the selected version, and if so, selecting that different version as the selected version of the assembly for binding to the application program.

The Office action rejected claim 28 as unpatentable over Evans. Specifically, the Office action contends that Evans teaches several recitations of claim 28 and cites similar sections of Evans as cited previously with respect to claims 1 and 17. The Office action concedes that Evans does not teach if determining whether publisher configuration information is associated with the assembly, then interpreting information in the publisher configuration to determine whether to bind to the application program a version of the assembly that is different from the selected version. However, the Office action contends that the publisher configuration would be known information because a designer has first-hand knowledge of the criterion under which the application operates. Thus, the recitations of claim 28 would be rendered obvious. Applicants respectfully disagree, challenge the Office action's unsupported conclusions and respectfully request withdrawal of the §103(a) rejections of the claims, or specifically request that a reference or references, including the required motivation to combine, be provided demonstrating otherwise. See M.P.E.P. § 2144.03.

As discussed above, Evans does not teach assemblies of objects, let alone show any appreciation of the concept of using assemblies of objects to link to executable code as an assembly. Further, the publisher configuration referred to in claim 28 is associated with its corresponding assembly and may be interpreted to determine which version of the assembly to bind to the executable code. Evans merely teaches a single tiered system for determining which object version to link to executable code. Thus, Evans cannot possibly teach a method for determining whether a configuration is associated with an assembly of objects, interpreting the configuration, and binding a specific version of the assembly to executable code based on the interpreted configuration information.

For at least these reasons, applicants submit that claim 28, and dependent claims 29-32, by similar analysis, are patentable over the prior art of record.

Turning to the last independent claim, claim 33 recites a computer-implemented method, comprising, receiving a request corresponding to binding a selected version of an assembly to an application program, determining whether application configuration information is associated with the application, and if so, interpreting the application configuration information to determine whether to bind to the application program a version of the assembly that is different from the selected version, and if so, selecting that different version as the selected version of the assembly for binding to the application program, determining whether publisher configuration information is associated with the assembly, and if so, interpreting information in the publisher configuration to determine whether to bind to the application program a version of the assembly that is different from the selected version, and if so, selecting that different version as the selected version of the assembly for binding to the application program, and determining whether administrator configuration information exists, and if so, interpreting administrator configuration information to determine whether to bind to the application program a version of the assembly that is different from the selected version, and if so, selecting that different version as the selected version of the assembly for binding to the application program.

The Office action rejected claim 33 as unpatentable over Evans. Specifically, the Office action contends that Evans teaches several recitations of claim 33 and cites similar sections of Evans as cited previously with respect to claims 1 and 17. The Office action concedes that Evans does not teach if determining whether publisher configuration information is associated with the assembly, then interpreting information in the publisher configuration to determine whether to bind to the application program a version of the assembly that is different from the selected version. However, the Office action contends that the publisher configuration would be known information because a designer has first-hand knowledge of the criterion under which the application operates. Thus, the recitations of claim 33 would be rendered obvious. Applicants respectfully disagree, challenge the Office action's unsupported conclusions and respectfully request withdrawal of the §103(a) rejections of the claims, or specifically request that a reference or references, including the required motivation to combine, be provided demonstrating otherwise. See M.P.E.P. § 2144.03.

As previously discussed, Evans does not teach assemblies of objects, let alone show any appreciation of the concept of using assemblies of objects to link to executable code as an assembly. Further, the publisher configuration referred to in claim 33 is associated with its corresponding assembly and may be interpreted to determine which version of the assembly to bind to the executable code. Evans merely teaches a single tiered system for determining which object version to link to executable code. Thus, Evans cannot possibly teach a method for determining whether a configuration is associated with an assembly of objects, interpreting the configuration, and binding a specific version of the assembly to executable code based on the interpreted configuration information.

For at least these reasons, applicants submit that claim 33, and dependent claims 34-37, by similar analysis, are patentable over the prior art of record.

For at least these reasons, applicants submit that all the claims are patentable over the prior art of record. Reconsideration and withdrawal of the rejections in the Office action is respectfully requested and early allowance of this application is earnestly solicited.

CONCLUSION

In view of the foregoing remarks, it is respectfully submitted that claims 1-37 are patentable over the prior art of record, and that the application is good and proper form for allowance. A favorable action on the part of the Examiner is earnestly solicited.

If in the opinion of the Examiner a telephone conference would expedite the prosecution of the subject application, the Examiner is invited to call the undersigned attorney at (425) 836-3030.

Respectfully submitted,

[illegible]

Albert S. Michalik, Reg. No. 37,395
Attorney for Applicants
Law Offices of Albert S. Michalik, PLLC
704 - 228th Avenue NE, Suite 193
Sammamish, WA 98074
(425) 836-3030
(425) 836-8957 (facsimile)

CERTIFICATE OF MAILING

I hereby certify that this Amendment and Petition for Extension of Time, along with Transmittal are being deposited with the United States Postal Service on the date shown below with sufficient postage as First Class Mail in an envelope addressed to: Commissioner for Patents, Alexandria, VA 22313.

Date: April 12, 2004

Albert S. Michalik

Albert S. Michalik

2511 Amendment